NOT DONE - SHARE - FE Reqs

Monday, May 01, 2017 11:03 AM

info and general requirements/instructions for the 'votersrevenge-frontend' open source project

INTRO

github repo: https://github.com/sldev2/votersrevenge-frontend

this starting repo was made mostly by using vue ui. It has the following libraries, and the delivered project (ultimately to be delivered in a branch called 'develop' on this repo) should use the same ones (you can check with me if there's a need to use different versions)

```
 "dependencies": {
   "@vue/composition-api": "^1.1.1",
   "postcss": "^7",
   "tailwindcss": "npm:@tailwindcss/postcss7-compat@^2.0.2",
   "vue": "^2.6.11",
   "vue-router": "^3.2.0",
   "vuex": "^3.4.0"
 },
 "devDependencies": {
   "@storybook/vue": "6.0.26",
   "@types/jest": "^24.0.19",
   "@vue/test-utils": "^1.2.2",
   "eslint": "^6.7.2",
   "typescript": "~4.1.5",
   "vue-cli-plugin-storybook": "^2.1.0",
   "vue-cli-plugin-tailwind": "^2.0.6",
```

pnpm was used to install libraries. please continue using pnpm to install new libraries needed (not npm or yarn)

the results of this open source project will be used in another open source project called "votersrevenge", @ github.com/sldev2/votersrevenge. See votersrevenge.info for more information on the Voter's Revenge project.

A *posse* is groups of citizens; in the US wild west, posses used to hunt down criminals; the votersrevenge app, for which this project is being developed, has the purpose of empowering citizens to "fire" elected politicians who are corrupt, ineffective, etc. In the US, it is in general not possible for citizens to remove elected officials (though it should be…). Consequently, by "firing" a politician, I mean ensuring that the politician doesn't get re-elected. (In the case of the wrangler role, the idea is to exert pressure on elected officials by shaming the them, while educating fellow citizens about their misdeeds. This will indirectly damage their re-election prospects.)

A *local posse* is a subset of a posse. There will typically be many local posses for each posse (eventually). If a user of the system joins a local posse, directly, he/she is automatically made a member of the parent posse. However, joining a posse doesn't require joining a local posse. Local posse members will typically live near each other, while posse members, in general, will only live in the same state (potentially far from each other)

It is hoped that citizens across the world will find votersrevenge useful, including when dealing with parliamentary democracies, which the US is not.

The crux of the votersrevenge-frontend project is to allow visitors to to the votersrevenge website to

1) **become active members of posses (and child *local posses*) via a registration process that requires them to**
    a. **agree to at least one pledge - either a voteslinger or wrangler type of pledge**

   b. specify their (nearby, approximate) location with the help of a map and geolocation api
   c. make additional pledges, as time goes on, now including a shindig type of pledge

2) become follower type of members, which requires no pledge

3) scroll through collections of posses, before and after filtering these collections
   a. there will always be a filter applied by (US) state

4) scroll through a collections of all local posses belonging to the same parent posse

5) identify members who live nearby, by their user-names, for future contact through a messaging program (the messaging program is not in the scope of this project; it is to be housed at github.com/sldev2/votersrevenge-messaging)

INFORMATION

roles and registration

super_sheriff	registered
sheriff	registered
deputy	registered
voteslinger	registered
wrangler	registered
follower	registered
visitor	not registered

registered user can have different roles in different posses and any of their local posses, though a super_sheriff is a super_sheriff in all posses and all local posses

users can only browse through posses on the home page, not any of their local posses. a simple count of associated local posses is shown on the home page, as part of a link to a local posse page, which shows only local posses belonging to the parent posse displayed on the home page

users can only browse through local posses on pages other than the home page; such local posses will all belong to the same parent posse

REQUIREMENTS

- □ instructions on how to install and run whatever bits are delivered (feel free to ask if I already know how do something)
- □ the front end must access the database only through a GraphQL api, generated by the Hasura GraphQL engine, with MS SQL Server in the back end

- □ provide an Oauth service for a related app (which is the voter's revenge chat app)
    - IOW, registered users in this website will be able to use the votersrevenge-messaging website with this website's login credentials
- □ provide 'idiot instructions' on how to use these credentials at votersrevenge-messaging

MILESTONE #1

- □ create a schema for the database, with a database diagram
  // by "schema", I just mean tables and fields definitions; NOT any stored procs or triggers

MILESTONE #2

- □ create the database
    - Tables (THIS IS JUST A STARTING GUIDE you will need to add tables, fields, indexes, I think a stored proc or trigger or two // also, you may need to restructure the database, somewhat; you will generally add data for the next milestone // though the lookup tables can be populated, in this milestone):

        // generally, I think every table X will have an id field: X_id

        - □ user
            - □ user_id
            - □ username
            - □ email // nullable, system email
            - □ is_validated // got validation of the email during the registration process
            - □ public_email // nullable, visible to everybody
            - □ statement // nullable, 1200 char max

        - □ user_location
            - □ user_id // user : user_location is 1 to 0 , since limited users (i.e., 'followers') don't have to provide their location
            - □ state // nullable, 5 chars max

            // if a geolocation type object is sent over the wire, have db stored proc/trigger determine lat, lng, and write those
            // also, if lat, lng is sent over the wire, have db stored proc determine the geolocation value, and write it, also
            // my guess is that only lat, lng should be sent over the wire; but geolocation data type is useful for doing queries like
            // "find all users within X miles of user Y", where user Y's location is already in the database

            - □ geolocation : geography // geography is a SQL Server data type
                - ◆ // default is
            - □ lat : float // not null
            - □ lng : float // not null
            - □ zipcode: string(10 char max) // nullable
        - □ government_level // a lookup table
            - □ government_level_id
            - □ name: string (max 15 chars)
            - □ abbr2 : string*2
            - □ abbr3 : string*3

            <u>values in this table:</u>

1	National	NT	NAT
2	State	ST	STA
3	County/Town	CT	CTY

        - □ government_house // a lookup table
            - □ government_house_id
            - □ name: string (max 15 chars)
            - □ abbr2 : string*2
            - □ abbr3 : string*3

            <u>values in this table:</u>

1	House	HR	HOR
2	Senate	SN	SEN
3	Executive	EX	EXE
4	Other	OT	OTH

        - issue
            - □ issue_id
            - □ name : string (max 50 chars)
            - □ short_desc : string (max 320 chars)
            - □ long_desc: string (max 1600 chars)

            values in this table

1	Election Steal 2020	Widespread fraud in the 2020 Federal Elections occurred, for both the Presidency and Congress, with over 3,000 sworn affidavits to this effect.	Election Steal 2020 lorem ipsum
2	Medical Tyranny	The regulatory bureaucracy in the US (including CDC and FDA) is non-transparent and out-of-control. Additionally, state governors and school boards are abusing their powers, all in the name of public health. Frontline medical workers who	Medical Tyranny lorem

		choose not to get vaccinated are being terminated.	ipsum
3	Medicare for All	Americans vastly overpay for their healthcare, and a "Medicare for All" single payer model will greatly rectify this price gouging.	Medicare for All lorem ipsum

- state
  - state_id
  - name string: (max 12 chars) non-null
  - abbr2 string*2 non-null
  - values in this table; one row for each state

1	Alabama	AL
2	Alaska	AK
3	Arizona	AZ
ETC.		

- target   // these are politicians, who typically are holding office
  - target_id
  - first_name :string (max 20 chars) // non-null
  - last_name   :string (max 28 chars) // non-null
  - current_position : string (max 25 chars) // non-null
  - state_id   // FK
  - government_level_id // FK

- □ posse
  - posse_id
  - name : string // this is calculated ; the formula is simply target.first_name + ' ' + target.last_name + ' - ' + issue.name
  - issue_id // FK
  - target_id // FK
  - government_level_id : integer // 1 is National, 2 is State, 3 is County/Town
  - government_house_id   // 1 is House, 2 is Senate, 3 is Executive, 4 is Other
- □ posse_members   // posse : posse_members is 1 to Many
  //not 100% sure about this... since the same user can be in multiple posses
  - posse_id // FK
  - user_id   // FK
- □ local_posse
  - posse_id // posse : local_posse is 1 to Many
  - nickname: string(20 chars max) ; non-null; at least 7 chars
  - name : string // this is calculated ; the formula is (parent) posse.name + ' ' + nickname
  - zipcode : string *5   // I'm aware of the fact that users's zipcodes have a different definition
  - quick_find_code : string*8 // strangers can search by zip-##, where zip Is 5 chars
    - e.g., is 2 local posses are located in zip code 07052, their quick_find_code's could be:

- □ local_posse_member   // local_posse : local_posse_members is Many to Many
  - posse_id
  - user_id
- □ posse_roles
  - posse_role_id
  - name
  - is_registration_required

values in this table

1	super_sheriff	true
2	sheriff	true
3	deputy	true
4	voteslinger	true

5	wrangler	true
6	follower	true
7	visitor	false

- □ default_redline_deadline   // these are only defined by sheriff and deputies; not all redlines are default redlines
  - target_id // FK
  - issue_id   // FK
  - posse_id // FK
  - redline_date
  - redline : string (max 450 chars)
  - deadline_date
  - deadline : string(max 450 chars)
  - pledge_type : integer // 1 for Voteslinger 2 for Wrangler or 3 for Shindig
  - is_general_election : boolean // only meaningful when pledge_type = 1, i.e., voteslinger

- □ pledge

  // a user has a choice between pledging to accept a default redline/deadline pair, created by admins
  // or else pledging to accept their own redline/deadline pair, defined by themselves
  // the exception is voteslinger deadlines and deadline_dates
  // these are set in stone by SYSTEM, but will be enforced primarily via business rules in the program
  // still, the database insertion routine should check that, for a given target_id and pledge_type==1, a
  // voteslinger pledge's deadline & deadline date must exist in default_redline_deadline
  // and, secondly, that a user cannot have multiple voteslinger pledges with the same target_id and deadline_date
  - user_id // user to pledge is 1 to Many
  - default_redline_deadline_id //nullable

    // if default_redline_deadline_id is null then
    // personal_redline_date, personal_redline, personal_deadline_date, and personal_deadline
    // must all be non-null and vice versa
  - personal_redline_date   // nullable
  - personal_redline string(max 450 chars)   // nullable
  - personal_deadline_date // nullable
  - personal_deadline string(max 450 chars) // nullable
  - fullfilled   // boolean, default = false
  - fulfillment_date // nullable
  - pledge_type   // 1 for Voteslinger 2 for Wrangler or 3 for create_Shindig or 4 for attend_shindig
    // a shindig pledge
    - □ only users who are sheriffs or deputies in a local posse can create shindigs

MILESTONE #3

- □ create the api, and populate database with sufficient data to test, and also create tests for each of the following items
  (**probably any test techology would do, but please check with me, first**)
  exception to "populate database with sufficient data to test" - just deliver with data corresponding to
  - □ determine how many local posses exist for any given posse
  - □ determine how many *wrangler* members there are in any given posse via 2 separate methods
    - □ method #1 just sum the wranglers found in the posse, ignoring local posses membership, completely
    - □ method #2 count wranglers in the posse who have no presence in any of the posses' local posse
      - add to this the wranglers who are in at least 1 local posse, also, but take care not to double count such wranglers who have that role in more than 1 such local posse
      - I expect this to be so computationally expensive that it will serve more as a validation check for back end maintainers and testers
  - □ determine how many *voteslinger* members there are in any given posse via 2 separate methods
    - □ method #1 just sum the voteslingers found in the posse, ignoring local posses membership, completely
    - □ method #2 count voteslingers in the posse who have no presence in any of the posses' local posse
      - add to this the voteslingers who are in at least 1 local posse, also, but take care not to double count such votesllingers who have that role in more than 1 such local posse
      - I expect this to be so computationally expensive that it will serve more as a validation check for back end maintainers and testers
  - □ determine how many *follower* members there are in any given posse via 2 separate methods

  - □ method #1 just sum the followers found in the posse, ignoring local posses membership, completely
  - □ method #2 count followers in the posse who have no presence in any of the posses' local posse
    - add to this the followers who are in at least 1 local posse, also, but take care not to double count such followers who have that role in more than 1 such local posse
    - I expect this to be so computationally expensive that it will serve more as a validation check for back end maintainers and testers
- □ determine how many days until the next redline_date occurs, for any given user_id and posse_id
- □ determine how many days until the next deadline_date occurs, for any given user_id and posse_id
- □ determine how many days until the next redline_date occurs, for any given user_id and local_posse_id
- □ determine how many days until the next deadline_date occurs, for any given user_id and local_posse_id
- □ determine how many days until the next default *voteslinger* redline_date occurs, for any posse_id
  - use the default_redline_deadline table
- □ determine how many days until the next default *general election voteslinger* deadline_date occurs, for any user_id and posse_id
  - use the default_redline_deadline table
- □ determine how many days until the next default *non-general election voteslinger*_deadline date occurs, for any posse_id
  - use the default_redline_deadline table
- □ determine posses that a user is a member of, given their user_id
  - return a collection of { posse_id, posse.name }
- □ determine local posses that a user is a member of, given their user_id

- □ determine how many voteslinger members there are in an given local posse
- □ determine how many wrangler members there are in any given local posse
- □ determine how many follower members there are in any given local posse
- □ determine how many shindig events there are in any given local posse

- □ create sample data, sufficient for testing (note: you can use the attached excel file for groups of users who are in some of 2 posses with their 4 local posses, between them)
  - the attached data file is for users in posses that operate at government_level = 2 (state)
  - **don't** add data or tests for any posses that operate at the government_level = 3 (county/town) // Day 2
  - □ do add data and tests for posses that operate at government_level = 1 (national)
    - please pick actual incumbent politicians from 2 states to be targets, viz., Georgia and New Jersey
    - please add test data for both Senators, and 2 House of Representative Congress persons from both these states
    - please add test data corresponding to all 3 issues defined in the issue table, viz
      - Election Steal 2020
      - Medical Tyranny
      - Medicare for All
    - for each posse at the national government_level, please add 2 local posses (humorous names greatly appreciated!)
    - please add at least 4 users from each role to each posse and at least 2 users from each posse to each of its local posses
      - // this is close to what I did in the attached data file, for government_level = 2

- □ implement the following business rules on the server side, if this makes sense (will also implement in the GUI)
  // return an error indicating the nature of the problem, when this occurs
  a new wrangler pledges' fulfillment_date must be within 2 months of the redline_date
  a new shindig pledges' fullfillment_date must be within the next 3 months (starting TODAY)

in gui, separate out primary from general voteslinger pledges, so that can fill fulfillment_date, correctly
    state those dates, clearly
gui business rule for wrangler pledges: must be within 2 months of the redline_date
gui business rules for shindig pledges:
    must be within the next 3 months (starting TODAY)
    only required if there is already somebody within 10 miles
    honest attempts at shindigging count honorably fulfilling this pledge (your neighbors may not be neighborly)
    only applies to local posses

<table>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td><b>action</b></td><td><b>role or roles</b></td><td><b>specs (happy path, mostly)</b></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td><b>visitors</b></td><td></td></tr>
<tr><td>visits <b>home page</b></td><td></td><td>SYSTEM tries to use geolocation API, and figure out which US state that belongs to, and then <b>filters</b> the posses by that state.
If the SYSTEM can't determine the state that the user is in, or if there is any error attempting to do so, or else the location is not located in the US, then SYSTEM <b>selects</b> the 1st state, in alphabetical order that is found in the database, from the GUI (viz., the state filters dropdown) and <b>filter</b> the posses accordingly</td></tr>
<tr><td>follows a posse (thus becoming a follower)</td><td></td><td>SYSTEM displays <b>home page</b>, which is displaying information from a specific posse

visitor clicks "Follow this Posse"

SYSTEM checks if user is logged in; since not logged in, redirects to <b>login/sign up</b> page (or popup)

<table>
<tr><td>visitor logs in (implying he had previously registered)</td></tr>
<tr><td>OR</td></tr>
<tr><td>user clicks link to <b>sign up</b> page

SYSTEM displays <b>sign up</b> page(s)
<u><b>sign up</b> page 1</u>

<table>
<tr><td>username</td><td>req</td><td>string</td></tr>
<tr><td>password</td><td>req</td><td>encrypted string</td></tr>
<tr><td>email // private to system</td><td>req</td><td>emal string</td></tr>
<tr><td>public email // for networking</td><td>opt</td><td>email string</td></tr>
</table>

<u><b>sign up</b> page 2</u>
registration form should do geolocation with a map, in order to obtain latitude and longitude
1) first, SYSTEM asks for zip code which is used to center the map more precisely
2) if user does not provide it, SYSTEM informs them that that is fine, but only users who provide <b>approximate</b> location can join local posses (for security reasons, it's not desirable to provide an exact location) User is invited to be use a map on a pin to determine a more exact location, but to not place the pin exactly. E.g., in a typical suburb, place the pin within 1/4 to 2 miltes of your actual location

visitor registers

<table>
<tr><td>zipcode</td><td>opt</td><td>string - 5 chars</td></tr>
<tr><td>lat // latitude</td><td>opt</td><td>number</td></tr>
<tr><td>lng // longitude</td><td>opt</td><td>number</td></tr>
</table>

SYSTEM sends visitor an email verification</td></tr>
</table>
</td></tr>
</table>

visitor verifies email

SYSTEM completes registrations of visitor, with follower role in the posse they applied, through

follower (previous visitor) logs in, for the first time

SYSTEM displays logged in status

SYSTEM displays a welcome message

SYSTEM displays information:

"As a follower of {THIS POSSE}, you can view messages in the 'public announcements' channel" (hyperlink to the **message board**)
"In order to participate in the other message board channels of this posse, you will have to take an "action, action, action" pledge, and become either a wrangler or a voteslinger"

"More information: what is the difference between a wrangler and a voteslinger" (hyperlink to **wrangler vs votesinger page**)

joins a posse as either a voteslinger or wrangler

SYSTEM displays **home page**, which is displaying information from a specific posse

visitor clicks "Join this Posse"

SYSTEM checks if user is logged in; since not logged in, redirects to **login/sign up** page (or popup)

visitor logs in (implying he had previously registered)

OR

user clicks link to **sign up** page

SYSTEM displays **sign up** page(s)

sign up page 1

username	req	string
password	req	encrypted string
email // private to system	req	emal string
public email // for networking	opt	email string

sign up page 2

registration form should do geolocation with a map, in order to obtain latitude and longitude

1) first, SYSTEM asks for zip code which is used to center the map more precisely
2) if user does not provide it, SYSTEM informs them that that is fine, but only users who provide **approximate** location can join local posses (for security reasons, it's not desirable to provide an exact location) User is invited to be use a map on a pin to determine a more exact location, but to not place the pin exactly. E.g., in a typical suburb, place the pin within 1/4 to 2 miltes of your actual location

visitor registers

username	req	string
password	req	encrypted string

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>email   // private to system</td><td>req</td><td>emal string</td></tr>
<tr><td>public email // for networking</td><td>opt</td><td>email string</td></tr>
<tr><td>zipcode</td><td>opt</td><td>string - 5 chars</td></tr>
<tr><td>lat // latitude</td><td>opt</td><td>number</td></tr>
<tr><td>lng // longitude</td><td>opt</td><td>number</td></tr>
</table>

<b>pledge page</b> (<b>sign up</b> page 3)

see PLEDGE PAGE, below

SYSTEM sends visitor an email verification

visitor verifies email

SYSTEM completes registrations of visitor, with follower role in the posse they applied, through

follower (previous visitor) logs in, for the first time

SYSTEM displays logged in status

SYSTEM displays a welcome message

SYSTEM displays information:
"As a follower of {THIS POSSE}, you can view messages in the 'public announcements' channel" (hyperlink to the <b>message board</b>)
"In order to participate in the other message board channels of this posse, you will have to take an "action, action, action" pledge, and become either a wrangler or a voteslinger"

"More information: what is the difference between a wrangler and a voteslinger" (hyperlink to <b>wrangler vs votesinger page</b>)
</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td>followers +</td><td>i.e., every user role that has a login: follower, wranglers, voteslingers, deputies, sheriffs</td></tr>
<tr><td></td><td>active user follows a new posse</td><td>SYSTEM displays a "follow this posse" button when
a) the user is logged in
b) the user is not already participating in the posse, in any role

user clicks a "follow this posse" button

SYTEM displays a friendly message "Howdy, pardner! you are now following the posse {NAME OF POSEE}. Please check out local chapters of this posse."
local chapters links to the Local Posse page (for this posse)</td></tr>
<tr><td></td><td>active user follows a new local posse of a posse the</td><td>SYSTEM displays a "follow this local posse" button when
a) the user is logged in</td></tr>
</table>

<table>
<tr><td></td><td>user already belongs to</td><td>b) the user is not already participating in the local posse, in any role

user clicks a "follow this local posse" button

SYTEM displays a friendly message "Howdy, pardner! you are now following the local posse {NAME OF POSSE}. Please check out local chapters of this posse."
local chapters links to the Local Posse page (for this posse)</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td>active members: voteslingers and/or wranglers</td><td></td></tr>
<tr><td></td><td>visitor joins a posse</td><td>visitor clicks "Join this Posse"

SYSTEM checks if user is logged in; if not logged in, redirects to login page (or popup)

user logs in

SYSTEM redirects to "Join this Posse" page

SYSTEM displays PLEDGE COMMITMENT FORM

SYSTEM displays a form

user enters:
<table>
<tr><td>username</td><td>req</td><td>string</td></tr>
<tr><td>password</td><td>req</td><td>encrypted string</td></tr>
<tr><td>email   // private to system</td><td>req</td><td>emal string</td></tr>
<tr><td>public email // for networking</td><td>opt</td><td>email string</td></tr>
<tr><td>lat // latitude</td><td>req</td><td>number</td></tr>
<tr><td>lng // longitude</td><td>req</td><td>number</td></tr>
</table>
SYSTEM displays "Congratulations, pardner! Before you can become a member, though, we need you make your first pledge. Remember, "Action, action, action!" Just talking doesn't catch any varmints!

SYSTEM displays list of all redline/deadline actions, with voteslinger and wrangler actions grouped separately

NOTE: Sql Server has a spatial type (data type), but I don't know how available such things are in other databases</td></tr>
<tr><td></td><td>follower joins a posse</td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

	LOCAL ADMIN / sheriff	
	messaging	SHARE - Slack Clone Reqs
	create a posse ("super posse")	
	super_sheriff	select : target issue EITHER: initial wrangler redline date and demand + initial wrangler deadline date OR: initial voteslinger redline date and demand + initial voteslinger deadline date
	creating a local posse	
create a posse	sheriff (owner)	select : posse target issue (opt.) edition name (opt.) edition location (lat / lon) EITHER: initial wrangler redline date and demand initial wrangler deadline date OR: initial voteslinger redline date and demand initial
deputize a posse follower	sheriff	sheriff adds an active posse member to the list of deputies SYSTEM show updated list of posse deputies
	ADMIN/ super_user ("super_sheriff")	
	CRU a target (GUI)	ADMIN
	Delete a target (stored proc only)	ADMIN
	CRU a an issue	ADMIN
	Delte an issue (stored proc only)	ADMIN
	CRU a redline/deadline template item	ADMIN
	CRU a posse redline/deadline template	ADMIN

	item	
	CRU a local posse redline/deadline	ADMIN
	local edition posse ("buddy posse") messaging vs. various user roles	3 channels: public announcements / announcements / buddy / follower 6 user chat roles (ordered by increasing privilege level): visitor (not logged in) follower buddy (voteslinger + wrangler) // "posse members" deputy // "posse members" sheriff // "posse members" super_sheriff // "posse members"
view public message	visitors + ('+' means: "and up")	view all messages in the "public announcements" channel
view non-public messages	follower +	view all messages in the "announcements", "buddy", and "follower" channels
participate in non-pubic message	posse members	in the in the "announcements", "buddy", and "follower" channels: initiate message reply message

non-public message limits	posse members: 4 messages per channel per day
	deputy+ : no limit
		// friendly "you have reached your daily quota for message; remember "action, action, action" " when user tries to exceed that quota
		// friendly "{user} tried to send you another message, today, but had already exceeded their quota"
		// show quota status with "X of 4 max messages, today; remember "action, action, action" "

private messaging	posse members: 4 messages per 1-on-1 chat, per day	throttled; throttle details TODO
		must send 'buddy request' (like a friend request in facebook) first (and be accepted) can only send 1 'buddy request' to another user in the same posse every 6 months
		
		// friendly "you have reached your daily quota for message; remember "action, action, action" " when user tries to exceed that quota
		// friendly "{user} tried to send you another message, today, but had already exceeded their quota"
		// show quota status with "X of 4 max private messages with { user} , today; remember "action, action, action" "

PLEDGE PAGE EXPLANATION

Tuesday, August 17, 2021 9:57 PM

wrangler

- the basic idea is that in order to be a wrangler in a given posse (or local posse), you must take a wrangler pledge, which has 2 events associated with it. Both events have specific dates associated with them
  - a redline event (a demand which must be fulfilled by a politician); called "redline", for short
  - a payback event; called "deadline", for short
    - paybacks are punitive actions which occur only if the redline demand is not satisfied by the redline date
    - since the paybacks are conditional, in a sense the politician is forgiven….. for the time being!
    - while a payback will damage the re-election prospects of the politician, wranglers are not pledged to vote against the posse's target politician (unike a voteslinger; see below)
- redlines and payback deadlines always occur in pairs, with the redline preceeding the payback deadline

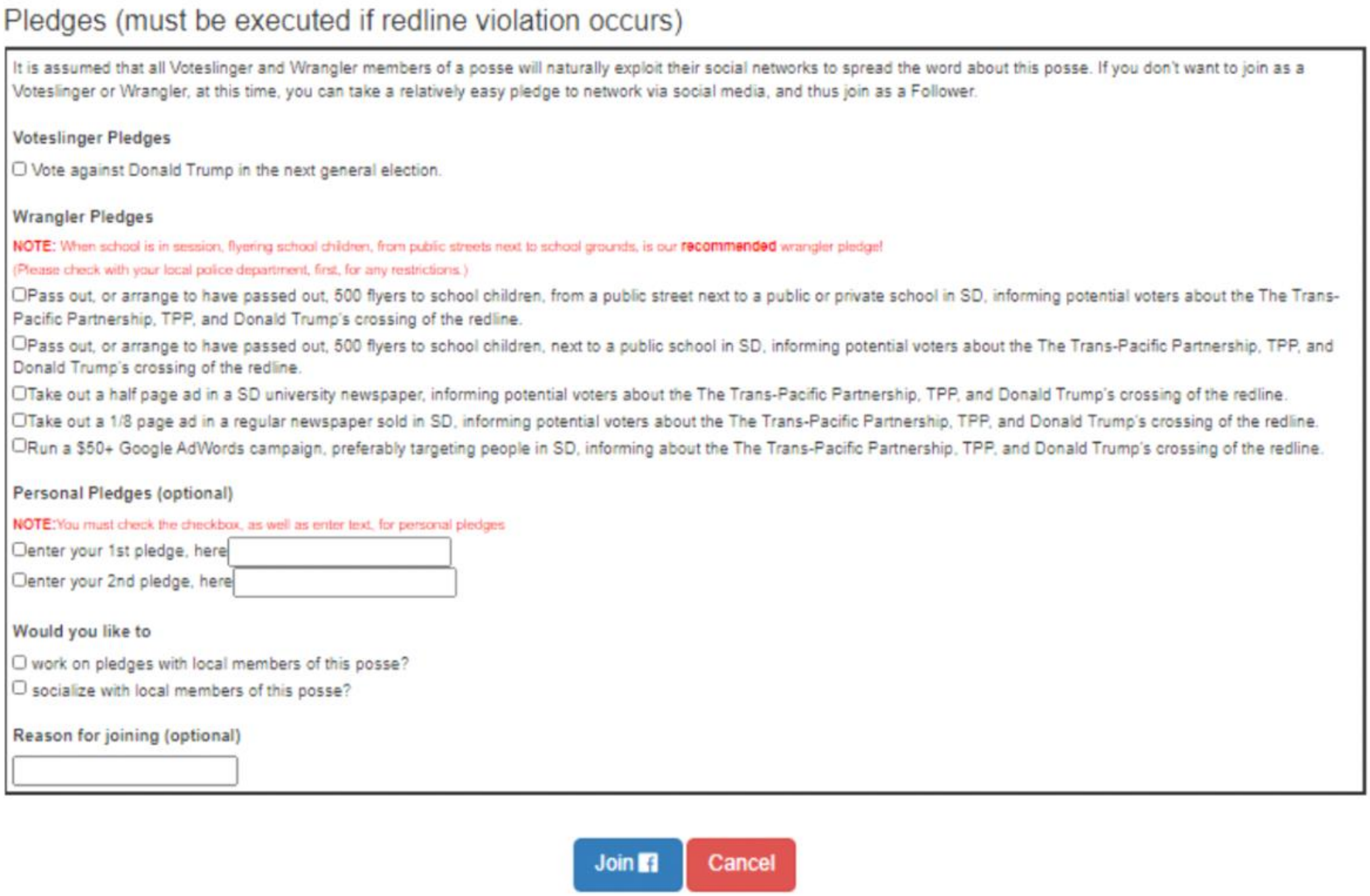
voteslinger

- the basic idea is that in order to be a voteslinger, you must take a voteslinger pledge, which has 2 events associated with it. Both have specific dates associated with them
  - a redline (a demand which must be fulfilled by a politician); called "redline" for short
  - an election deadline, **which is always associated with a vote** against the politician; called "deadline" for short
    - paybacks are punitive votes (i.e., a vote for a competitor to the posse target politician) which occur only if the redline demand is not satisfied by the redline date
    - since a violated redlines means that the voteslinger is 100% pledged to vote against the politician, there is no point in taking additional voteslinger pledges related to the same deadline vote; you cannot vote against somebody, on the same date, more than once! Thus, a voteslinger pledge is terminal, and unforgiving. The only forgiveness that might occur within the same election cycle, is if the voteslinger is pledged to vote *against* a politician in a *primary*, but a voteslinger redline related to a general election vote is *not* violated, freeing the voteslinger to vote *for* the politician in his *general* election
- redlines and election deadlines always occur in pairs, with the redline preceeding the election deadline

additionally

- within the same posse, or within the same local posse, a user can be both a wrangler and a voteslinger
- it is highly recommended that all voteslingers continue taking on new wrangler pledges until Primary Day and Election Day, even if their voteslinger redline is violated
- it is expected that early on, and in the middle part, of a politician's term of office, wranglers will be more effective than voteslingers, as the politician will hope and expect that committed voters will forget their anger on Election Day, and that most of the public will not be paying close attention (unless a wrangler payback action commands their attention)
- it is expected that, during the latter part of a politician's term of office, voteslingers will be more effective than wranglers

'classical' pledge page example (DO NOT consider this part of the spec; it's just to give an idea, though there's a lot of overlap with the spec; note that the layout is bad, with checkboxes right up against text)



Pledges (must be executed if redline violation occurs)

It is assumed that all Voteslinger and Wrangler members of a posse will naturally exploit their social networks to spread the word about this posse. If you don't want to join as a Voteslinger or Wrangler, at this time, you can take a relatively easy pledge to network via social media, and thus join as a Follower.

Voteslinger Pledges

☐ Vote against Donald Trump in the next general election.

Wrangler Pledges

NOTE: When school is in session, flyering school children, from public streets next to school grounds, is our **recommended** wrangler pledge!

(Please check with your local police department, first, for any restrictions.)

☐Pass out, or arrange to have passed out, 500 flyers to school children, from a public street next to a public or private school in SD, informing potential voters about the The Trans-Pacific Partnership, TPP, and Donald Trump's crossing of the redline.

☐Pass out, or arrange to have passed out, 500 flyers to school children, next to a public school in SD, informing potential voters about the The Trans-Pacific Partnership, TPP, and Donald Trump's crossing of the redline.

☐Take out a half page ad in a SD university newspaper, informing potential voters about the The Trans-Pacific Partnership, TPP, and Donald Trump's crossing of the redline.

☐Take out a 1/8 page ad in a regular newspaper sold in SD, informing potential voters about the The Trans-Pacific Partnership, TPP, and Donald Trump's crossing of the redline.

☐Run a $50+ Google AdWords campaign, preferably targeting people in SD, informing about the The Trans-Pacific Partnership, TPP, and Donald Trump's crossing of the redline.

Personal Pledges (optional)

NOTE: You must check the checkbox, as well as enter text, for personal pledges

☐enter your 1st pledge, here

☐enter your 2nd pledge, here

Would you like to

☐ work on pledges with local members of this posse?

☐ socialize with local members of this posse?

Reason for joining (optional)